

# Automatic Calibration of Large Traffic Models

Félix Mézière\* Gabriel Gomes\*\*

California PATH Program  
University of California Berkeley

**Abstract**—Some abstract

**Index Terms**—Large traffic, calibration, imputation, optimization, black box, evolutionary algorithm, freeway model, CTM.

## INTRODUCTION

SOME introduction

- What problem we solve
- History of the problem
- Our method and what it contributes
- What is a large traffic model, what is it for, why calibrate it.
- Explain shortly how we are gonna calibrate, mention *knobs, templates, fitness function to minimize, performance calculators, fitting real data*
- This work is mostly empirical/experimental, it is a base.
- Announce in what order we are going to proceed ('first blablabla, then blablabla ... finally blablabla).

we should probably insist from the beginning on the congestion pattern matching, which is the main feature.

## I. MODELING AND NOTATION

### A. Freeway model

We consider a one-way segment of freeway and its on-ramps and off-ramps. A fraction of the links is monitored. Define without notation:

- topography of the freeway
- scenario
- links
- lanes
- nodes
- mainline
- on-ramp
- off-ramp
- ramps
- source
- sink
- linear order
- flow
- density
- demand
- monitored link

- other stuff that I've not thought about

*Freeway model assumptions:* Describe here the assumptions we've made on the freeway model:

- the entrance link is monitored
- the behavior of the HOV
- other assumptions I've not thought about

### B. Traffic model

The *large traffic model* to calibrate is characterized by the following inputs and outputs:

- *Input:*
  - duration of the scenario and time step.
  - value of the exit flow demand at every source and non-mainline sink for every time step (total over the time step).
  - other parameters proper to the model, assumed to be already calibrated
- *Output:*
  - value of the exit flow on every link, for every time step (total over the time step).
  - value of the density on every link, for every time step (average over the time step).

*Traffic model assumptions:* Describe here the assumptions we've made on the traffic model:

- $\approx 0$  cars beginning and end of the day
- $\approx$ no queue on the ramps (at least on the off-ramps). Partially justified by the fact that we only input flows that are inferior to the capacity of each ramp (physical box constraints).
- other assumptions I've not thought about

### C. Data

We assume that we are in possession of the following measurements on the freeway:

- Value of the exit flow on some of the mainlines links, for every time step.
- Value of the density on the same mainline links, for every time step.
- Value of the exit flow on some of the ramps, for every time step.

*emphData assumptions:* Describe here the assumptions we've made on the traffic model:

- The first and last mainline link (entrance and exit of the freeway portion) are monitored.

\*Ecole polytechnique [felix.meziere@polytechnique.edu]

\*\*University of California Berkeley [gomes@path.berkeley.edu]

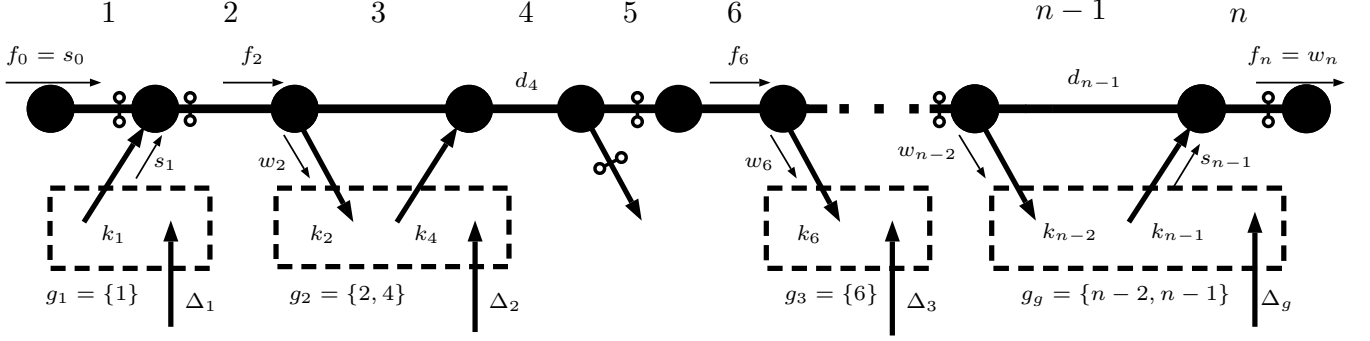


Fig. 1. Freeway model and notation

#### D. Notation

1) *Freeway model*: Let  $M = \llbracket 1, n \rrbracket$  the set of *mainline link* indexes,  $R \subset M$  the set of mainline link indexes whose exit index is connected to a ramp (i.e. the set of ramp indexes).

We denote as  $T \subset M$  the set of monitored mainline links (T for 'tracked') and  $K = \{i_1, i_2, \dots, i_\kappa\} \subset R$  the set of non-monitored ramps (K for "knobs").

$(L_i)_{i \in M}$  are the lengths of the mainline links.

2) *Traffic model*: Let  $dt$  the time step of the model and  $D$  its duration. There is a total of  $\frac{D}{dt} + 1$  time steps in the scenario. We will denote the set of these times steps  $\tau = dt \cdot \llbracket 0, \frac{D}{dt} \rrbracket$

$\forall t \in \tau$ ,  $(f_i(t))_{i \in M}$  and  $(r_i(t))_{i \in R}$  are the flows exiting respectively the mainline and ramp links at time  $t$ , output by the model.

In addition,  $f_0(t)$  is the flow entering the first mainline link (entrance of the freeway).

$\forall t \in \tau$ ,  $(d_i(t))_{i \in M}$  are the densities on the mainline links, output by the model.

$\forall t \in \tau$ ,  $\{\tilde{f}_0(t); (\tilde{r}_i(t))_{i \in M}\}$  are the exit flow demands imputed to the model.

3) *Data*: As a common pattern, the measured values will be denoted with a tilde.

Measured mainline exit flows are denoted  $(\tilde{f}_i(t))_{i \in T}$

Measured mainline densities are denoted  $(\tilde{d}_i(t))_{i \in T}$

Measured ramp exit flows are denoted  $(\tilde{r}_i(t))_{i \in T}$ .

Fig. 1 summarizes the model. The missing notation will be introduced progressively.

## II. UNCERTAINTY

Our problem involves three sources of uncertainty:

- Uncertainty on the data: the measurements have a certain confidence interval.
- Uncertainty due to the inexactness of the model itself. This uncertainty reflects the fact that, even if we had perfect data and the demand on every link, the

model would not output the exact real traffic (and congestion phenomena etc.).

- Uncertainty due to the inexact shape of the templates.

<- We gotta decide how much we develop this

These uncertainties are merged into two uncertainties:

- *Uncertainty on the local duration-long measurements*: This describes the uncertainty at the link level. It is applied to the sum of the flow measurements of one sensor for all the duration.

Denoting  $F_i = \sum_{t \in \tau} f_i(t)$  and  $\tilde{F}_i = \sum_{t \in \tau} \tilde{f}_i(t)$ , this local uncertainty is divided into two competing components:

- *additive local uncertainty*: denoted  $U^{add}$ . The additive confidence interval for  $F_i$  is:

$$F_i \in [\tilde{F}_i - U^{add}, \tilde{F}_i + U^{add}]$$

- *multiplicative local uncertainty*: denoted  $U^{mul}$ . The multiplicative confidence interval for  $F_i$  is:

$$F_i \in [\tilde{F}_i \cdot (1 - U^{mul}), \tilde{F}_i \cdot (1 + U^{mul})]$$

- *Uncertainty on the global duration-long measurements*: This describes the uncertainty at the whole mainline level. It is a generic multiplicative uncertainty applied to all quantities that are computed from the measurements on every mainline sensor during the whole duration.

We denote this uncertainty  $U^{global}$ . Let  $\tilde{q}_i(t)$  a quantity computed from the measurements on link  $i$  at time  $t$ . Denoting  $Q = \sum_{i \in T} \sum_{t \in \tau} q_i(t)$  and

$\tilde{Q} = \sum_{i \in T} \sum_{t \in \tau} \tilde{q}_i(t)$ , the global confidence interval for this quantity is:

$$Q \in [\tilde{Q} \cdot (1 - U^{global}), \tilde{Q} \cdot (1 + U^{global})]$$

<- We've got to decide how much we justify this definition of the uncertainties

## III. PROBLEM FORMULATION

### A. Introduction

For every monitored source or sink (except the mainline sink), we input the measured flow to the model as exit

flow demand.

The assumptions made in I-B imply that this demand is approximately equal to the actual flow going through the ramp, for all times and above mentioned links:

$$\forall i \in R \setminus K, \forall t \in \tau, \tilde{r}_i(t) = \bar{r}_i(t) \approx r_i(t) \quad (1)$$

The only missing parameters to our model are the flow demand profiles of the non monitored ramps:  $(\bar{r}_i(t))_{i \in K, t \in \tau}$ . Our method consists in mapping these  $\kappa$  flow profiles into one parameter each.

To do that, a flow profile called *template* is built for every non-monitored ramp. These templates, denoted  $(t_i(t))_{i \in K, t \in \tau}$ , consist in a normalized flow profile: a flow value is given to each element of  $\tau$  ( $dt$  during  $D$ ), and the resulting profile is normalized to a reasonable value  $T$ . For each of the non-monitored ramps  $i$ , we define a multiplicative factor  $k_i$  called *knob* that will set the intensity of the template. That is, we input as exit flow demand of the ramp its corresponding template multiplied by the ramp knob :  $k_i \cdot t_i(t)$ .

The parameters of our imputation problem are therefore the  $\kappa$  **knobs**, corresponding to the  $\kappa$  non-monitored ramps.

In addition, due the same assumptions that gave Eq. 1, we have :

$$\forall i \in K, \forall t \in \tau, k_i \cdot t_i(t) = \bar{r}_i(t) \approx r_i(t)$$

and, especially, with  $\Theta = \sum_{t \in \tau} t_i(t)$  :

$$\sum_{t \in \tau} r_i(t) = k_i \cdot \Theta \quad (2)$$

### B. Constraints on the parameters

We define here the constraints verified by the knobs. They consist in box hard boundaries and linear inequalities.

*Notation:*

$\vec{k} = (k_{i_1}, k_{i_2}, \dots, k_{i_\kappa})$  is the vector containing the values of the knobs.

$\sigma = (\sigma_{i_1}, \sigma_{i_2}, \dots, \sigma_{i_\kappa})$  is the source/sink indicator vector for the knobs:

$$\forall j \in K, \sigma_j = \begin{cases} 1 & \text{if ramp } j \text{ is an on - ramp} \\ -1 & \text{if ramp } j \text{ is an off - ramp} \end{cases}$$

For clarity, we will often abusively use the expression *knob-ramp*  $i$  instead of *ramp corresponding to the knob*  $i$ .

1) *Physical boundaries:* The box constraints applied to each knob are physical capacity limits imposed by the ramp they are associated with. They reflect that the maximum value of the ramp flow cannot exceed the capacity of the ramp.

$\forall i \in K$ , the maximum  $m_i$  of knob  $i$  is defined by:

$$\forall t \in \tau, k_i \cdot t_i(t) \bar{r}_i(t) \leq [\text{Capacity of the ramp associated to knob } i]$$

$$\Rightarrow m_i = \frac{[\text{Capacity of the ramp associated to knob } i]}{\max_t t_i(t)}$$

therefore :

$$\forall i \in K, \forall p \in \mathbb{N}, 0 \leq k_i \leq m_i$$

$$\Leftrightarrow \vec{0} \leq \vec{k} \leq \vec{m}, \text{ with } \vec{m} = \begin{bmatrix} m_1 \\ m_2 \\ \vdots \\ m_\kappa \end{bmatrix}$$

Define  $\leq$  for vectors ?

Define these boundaries as a box  $B$  ?

2) *Knob groups and flow balance:* We define here objects and notation to describe a simple situation: the knobs are closely monitored by nearby mainline sensors, leading often to a situation where a ramp is the only non-monitored ramp between two mainline sensors. This is equivalent to it being monitored, if it was not for the uncertainties.

We call *segment* the set of links between two consecutive mainline sensors, including the links containing these sensors.

We call *knob group* each set of knobs whose corresponding ramp is connected to the same monitored segment. This definition is illustrated in Fig. 1.

We call *partially monitored segment* the monitored segments associated with a knob group i.e. containing at least one non-monitored ramp.

Denoting  $\gamma$  the total number of knob groups, we have:

$$\text{Partially monitored segments : } (S_i)_{i \in \llbracket 1, \gamma \rrbracket}$$

Formal definition:

$$\exists ! \gamma \in M, \exists ! ((\beta_i, \eta_i))_{i \in \llbracket 1, \gamma \rrbracket} \in (T^2)^{\llbracket 1, \gamma \rrbracket} \text{ s.t.,}$$

denoting  $S_i = \llbracket \beta_i, \eta_i \rrbracket$  and  $S = \bigcup_{i \in \llbracket 1, \gamma \rrbracket} S_i$  :

$$\forall i \in \llbracket 1, \gamma \rrbracket,$$

- 1)  $\llbracket \beta_i, \eta_i \rrbracket \cap T = \{\beta_i, \eta_i\}$
- 2)  $\llbracket \beta_i, \eta_i \rrbracket \cap K \neq \{\emptyset\}$

$$\text{and } \forall k \in K, k \in S.$$

and

$$\text{Knob groups : } (g_i)_{i \in \llbracket 1, \gamma \rrbracket}$$

Formal definition:

$$\forall i \in \llbracket 1, \gamma \rrbracket, g_i = \llbracket \beta_i, \eta_i \rrbracket \cap K$$

We can deduce the value of the daily flow brought

by the knobs of each group from the *knob group flow balance*: the difference between all the flows entering and all the flows exiting their incomplete monitored segment. That is the sum of the flow exiting the mainline entrance of the segment and the flows exiting the monitored on-ramps throughout the segment minus the sum of the flow exiting the mainline exit of the segment and the flows exiting the monitored off-ramps throughout the segment.

**Put here the paragraph Gabriel wrote**

*Knob group flow balances* :  $(\Delta_i)_{i \in \llbracket 1, \gamma \rrbracket}$

Formal definition:

$\forall i \in \llbracket 1, \gamma \rrbracket, \forall t \in \tau,$

$$\Delta_i = \sum_{t \in \tau} \left[ \tilde{f}_{\beta_i}(t) - \tilde{f}_{\eta_i}(t) + \sum_{j \in (R \setminus K) \cap S_i} \sigma_i \cdot \tilde{r}_j(t) \right]$$

The balance equation of each partially monitored segment is:

$$0 = \sum_{t \in \tau} \left[ f_{\beta_i}(t) - f_{\eta_i}(t) + \sum_{j \in R \cap S_i} \sigma_i \cdot r_j(t) \right]$$

As stated in eq. 1, the model-output flows exiting the ramps are equal to their demand flows. The balance equation becomes therefore:

$$0 = \Delta_i + \sum_{t \in \tau} \left[ \sum_{j \in K \cap S_i} \sigma_i \cdot r_j(t) \right]$$

Leading to:

$$\begin{aligned} \Delta_i &= - \sum_{t \in \tau} \left[ \sum_{j \in K \cap S_i} r_j(t) \right] \\ \Leftrightarrow \Delta_i &= - \sum_{j \in K \cap S_i} \sigma_i \left[ \sum_{t \in \tau} r_j(t) \right] \end{aligned}$$

and thanks to eq. 2:

$$\Leftrightarrow \Delta_i = - \sum_{j \in K \cap S_i} \sigma_j \cdot k_j \cdot \Theta \quad (3)$$

Eq. 3 shows that, for every knob group, the knobs composing it are linked by one linear equation.

This equation determines uniquely the value of the single-knob groups and links the multiple-knob groups with one linear constraint. The next paragraph describes how we apply uncertainties to this equation in order to produce new, closer to reality knob boundaries.

3) *Refined knob boundaries*: The local uncertainty described in II prevents us from keeping Eq. 3 as a constraint for the parameters.

$\forall i \in \llbracket 1, \gamma \rrbracket$ , let the most permissive uncertainties :

$$\begin{aligned} \Delta_i^- &= \max \{ |\Delta_i| - U^{add}, |\Delta_i| \cdot (1 - U^{mul}) \} \\ \Delta_i^+ &= \max \{ |\Delta_i| + U^{add}, |\Delta_i| \cdot (1 + U^{mul}) \} \end{aligned}$$

Taking the local uncertainty into account in Eq. 3 is

translated into the following linear inequality constraint:

$$\forall i \in \llbracket 1, \gamma \rrbracket, \Delta_i^- \leq \left| \sum_{j \in K \cap S_i} \sigma_j \cdot k_j \cdot T_j \right| \leq \Delta_i^+ \quad (4)$$

*Comments:*

We can now illustrate and justify the form that we have adopted for the uncertainty. This method allows us to quantify the freedom given to the result: the flow balance of each knob-groups is between  $(1 - U^{mul})$  and  $(1 + U^{mul})$  times what has been measured by the mainline sensors, acknowledging that we don't accept less than  $\pm U^{add}$  cars precision on the measures. Taking into account  $U^{add}$  is indispensable. This is observed in the case of single-knob groups. In this case, Eq. 3 leads to a unique value for the knob of the group. Let us call it *perfect value of the knob i*, denoted  $k_i^*$ . It immediately follows from 4 that two new boundaries are set for  $k_i$ , if they are tighter than  $[0, m_i]$ . The computation of the perfect value leads sometimes to ridiculously small quantities. In these cases, the maximum obtained with  $1 \pm U^{mul} \cdot k_i^*$  corresponds often to a total daily flow of less than 50 cars exiting the ramp, which is not acceptable.

*Example:* One of the ramps has a perfect value of 0.02, which leads to a maximum of ?? cars going through the ramp during the whole day if  $U^{mul}$  is set to 100% (very permissive: the daily flow can double what is measured by the mainline sensors). This is too small for any scenario.

The fluctuation allowed by the new boundaries of this ramp is of ?? cars, which is ?? times smaller than  $U^{add}$ : the sensors do not have this level of precision, and the sensor noise/bias is responsible for this impossible perfect value.

Once  $U^{add} = [10\% \cdot (\text{measured daily mainline flow})]$  is taken into account, the maximum of the knob becomes 0.7, which corresponds to ?? cars and is in an acceptable range.

**THIS IS WHERE IVE STOPPED WORKING TODAY**

### C. Performance metrics and error calculators

Three performance metrics are used on the model output to measure the state of the freeway. Each of these metrics is then compared by an error calculator to its value computed on the data. The errors are expressed as a percentage, in order to monitor easily their relative importance. As exposed just below, the common principle for this error computation is to calculate the relative difference between the model output performance and the data performance.

1) *Vehicle Hours Travelled (VHT)*: This quantity is the sum of the time spent on the mainline by each car, over the whole duration. Obviously, it is computed using only the monitored mainline links, for the comparison with the data to be relevant. *explain why it is important in traffic study, why we have chosen it*

VHT computation on monitored mainline links output and data:

$$VHT(\vec{k}) = \frac{dt}{[1 \text{ hour}]} \sum_{i \in T} L_i \sum_{t \in \tau} d_i(t)$$

Denoting  $\widetilde{VHT}$  the value computed from the data using the same formula, the error is the relative difference :

$$E_{VHT}(\vec{k}) = \frac{|VHT(\vec{k}) - \widetilde{VHT}|}{\widetilde{VHT}}$$

2) *Vehicle Miles Travelled (VMT)*: This quantity is the sum of the distance traveled on the mainline by each car, over the whole duration. Obviously, it is computed using only the monitored mainline links, for the comparison with the data to be relevant. *Explain why it is important in traffic study, why we have chosen it*  
VMT computation on monitored mainline links output and data:

$$VMT(\vec{k}) = \sum_{i \in T} L_i \sum_{t \in \tau} f_i(t)$$

Denoting  $\widetilde{VMT}$  the value computed from the data using the same formula, the error is the relative difference :

$$E_{VMT}(\vec{k}) = \frac{|VMT(\vec{k}) - \widetilde{VMT}|}{\widetilde{VMT}}$$

*Reduction of the feasible space*: we present here a method used to reduce the feasible space size by forcing the knobs to match the correct VMT value.

VMT is the result of a simple *a priori* calculation that does not need the traffic model output calculation, if we know the boundary conditions at  $t = 0$  and  $t = D$ . As exposed in ??, we assume that these conditions are 0 cars on every link ( $D$  as to be big enough for these conditions to be very small in comparison with the total number of vehicles during  $D$ ).

Let  $VMT^{apriori}(\vec{k})$  the expected VMT value computed from  $\vec{k}$ .

Let  $VMT^{ref} = VMT(\vec{k}^{ref})$ , a certain VMT reference value output by the model. We suppose that  $\vec{k}^{ref}$  is some feasible knobs vector (in our case, we used  $\vec{k}^{ref} = (1, \dots, 1)$ ).

$$VMT^{apriori}(\vec{k}) = VMT^{ref} + \sum_{i \in K} \left[ \sigma_i \cdot k_i \cdot \Theta \cdot \sum_{\substack{j \in T \\ j > i}} L_j \right] \quad (5)$$

The *a-priori* calculation above relies on anticipating the changes from  $VMT^{ref}$  caused by changing the knobs from  $\vec{k}^{ref}$  to  $\vec{k}$ . For each knob, the flow change resulting from its modification is multiplied by the remaining mainline length and the "on/off-ramp indicator". All these contributions are then summed.

The linear equation 5 empowers us to constrain the input  $\vec{k}$  in order to ensure  $VMT(\vec{k}) \approx \widetilde{VMT}$ , thus reducing the size of the feasible space by one dimension.

3) *Congestion Pattern*: We call *contour plot* the graph representing the value of a quantity on every mainline link at all times: the mainline links as absciss and time steps as ordinates. Below is an example of a density contour plot for one day on a 135 links freeway (time is descending). This plot is used to monitor easily where and when the congestion is : here, it is contained in the framed part. In what follows, by analogy, we will call *contour domain* the set  $\mathcal{P} = \{(i, t) \mid i \in M, t \in \tau\}$  and *pixel* each of its elements.

On the model output, we define the *congested* pixels as the ones where the density exceeds some *critical density* deduced for each link from the freeway and traffic models. The main feature of our calibration method is to fit the locations and times of these congested pixels to what the measurements indicate.

For each mainline link  $i \in M$ , we denote  $d_i^*$  the critical density.

We define the *output congested domain*  $\mathcal{C} \in \mathcal{P}$  containing the congested pixels :

$$\mathcal{C} = \{(i, t) \in \mathcal{P} \mid d_i(t) \geq d_i^*\}$$

To define an error calculator based on  $\mathcal{C}$ , a domain supposed to contain the congestion as to be defined. From the data *density contour plot* (partial, obtained only on the monitored links), we define a domain  $\tilde{\mathcal{C}} \subset \mathcal{P}$  fitting the congested pixels as best as it can following some criteria (how this domain is built depends on the amount of data the operator possesses and on his goals. In our case,  $\tilde{\mathcal{C}}$  was a set of rectangles containing all the congestion seen in the data contour plot).

We can now define the congestion pattern error calculator denoted  $E_{CP}$  as the normalized number of wrong congestion state pixels. That is, we add one to  $E_{CP}$  for each pixel that is not congested but should and for each pixel that is congested but shouldn't. We then divide this result by the number of pixels that should be congested (i.e.  $Card(\tilde{\mathcal{C}}) = \sum_{t \in \tau} \sum_{i \in M} \mathbb{1}_{(i, t) \in \tilde{\mathcal{C}}}$ ).

$$E_{CP}(\vec{k}) = \frac{\sum_{t \in \tau} \sum_{i \in M} \mathbb{1}_{\{(i, t) \in (\tilde{\mathcal{C}} \setminus \mathcal{C}) \cup (\mathcal{C} \setminus \tilde{\mathcal{C}})\}}}{\sum_{t \in \tau} \sum_{i \in M} \mathbb{1}_{(i, t) \in \tilde{\mathcal{C}}}}$$

Note that this error becomes very sensible if the data does not contain much congestion ( $Card(\tilde{\mathcal{C}}) \ll Card(\mathcal{P})$ ).

#### D. Objective function

The fitness function to minimize,  $\Phi$ , is the weighted sum of 4 component functions: 3 error calculators, each coming from a different performance calculator (see ??), and the penalization mentioned in ??.



We define here these components as functions of a knobs values vector.

We normalize each component function in percentage following a criteria that is proper to it (see next 4 sub-sections). The common principle is to compute a relative difference using the PeMS data as reference.

**Still didn't found an elegant way to note & index the components / PCs**

$\Pi_2$ : Relative difference of Vehicle Miles Travelled on monitored mainline links, comparing BeATS output to PeMS data.

$\Pi_3$ : Relative difference of Vehicle Hours Travelled on monitored mainline links, comparing BeATS output to PeMS data.

$\Pi_4$ : Normalized number of links & times with the wrong congestion state (CP is for *congestion pattern*).

Let  $(\Pi_i)_{i \in \llbracket 1,4 \rrbracket}$  the components of the error function and  $\mathcal{Q}$  the search space.

$$\forall i \in \llbracket 1,4 \rrbracket, \Pi_i : \begin{cases} \mathcal{Q} & \longrightarrow [0,100] \\ \vec{k}^{(p)} & \longmapsto \Pi_i(\vec{k}^{(p)}) \end{cases}$$

**Should we define the PCs as function of p instead of  $\vec{k}^{(p)}$  ?**

$I^{global}$ , described in I-C, defines a tolerance threshold for the error calculator results: the error results below  $I^{global}$  are set to zero, in order to avoid any differentiation between them (we do not have a level of precision below  $I^{global}$ ). We call *contributions* the final contribution of every performance calculator in the sum composing the fitness function: the value in percentage multiplied by its weight, once the preceding tolerance has been applied.

Let  $(\rho_i)_{i \in \llbracket 1,4 \rrbracket}$  the weights.

We have :  $\sum_{i=1}^4 \rho_i = 1$  and  $\forall i \in \llbracket 1,4 \rrbracket, \rho_i > 0$

$$\Phi : \begin{cases} \mathcal{Q} & \longrightarrow [0,100] \\ \vec{k}^{(p)} & \longmapsto \sum_{i=1}^4 \rho_i \cdot \Pi_i(\vec{k}^{(p)}) \cdot \mathbb{1}_{(i=1) \cup (\Pi_i(\vec{k}^{(p)}) > I^{global})} \end{cases}$$

This definition as a weighted sum of percentages implies that the values of  $\Phi$  can be interpreted as a *global error percentage*. The weight given to each component is equivalent to the importance we want to give to each one of them. In fact, due to the very good performance of CMA-ES giving a significant weight to each one is enough to ensure that the result will be the best possible (**<- this is bullshit**).

Below is a description of the computation of each component.

1) *Optimization problem statement*: enter statement here

#### IV. NUMERICAL METHOD

##### A. Requirements

**Keywords and ideas:** *non-convex, rugged search landscape (i.e.: sharp bends, discontinuities, outliers, noise, local optima), badly scaled and/or highly non-separable, convergence, adaptive, Black-Box, evaluated search points are the only accessible information on f, Randomized/stochastic works well, Robust, imputation, continuous domain.*

- The performance calculator errors are irregular functions. In particular, the congestion pattern fitting reflects congestion phenomena. These present numerous thresholds in their non-smooth behavior.
- Another component of the error function is a distance involving a randomly generated point: derivative methods can't be applied to stochastic functions. We can also point out that the performance calculators described in ?? aren't always correlated.
- Furthermore, each evaluation of the error function requires the execution of a simulation (around 5 seconds on a desktop computer), and this evaluation is the only thing accessible of f: there is no way of quickly computing its value or its gradient.
- Therefore, the fitness function is a black box (it is the weighted sum of the performance calculators errors). We deduce from these observations that convex optimization methods and derivative-based methods are not adapted to our case.
- The search space is a continuous hypercube, as explained in V-E.
- We can conclude that we study a non-linear, non-convex black-box imputation problem in continuous domain. (**<- rewrite all this**)
- *Plus, the algorithm has to be adaptive, few parameters to tune if possible, no prior optimization knowledge required if possible*
- What kind of algorithm is suitable
- Why we only wanted one that works instead of testing several. Time is not a constraint. Quality is the goal.

##### B. Co-variance Matrix Adaptation - Evolution Strategy (CMA-ES)

Talk about CMAES characteristics and why it is very well suited for this.

**Most of the stuff will come from :**

- <https://www.lri.fr/hansen/cmaesintro.html>
- <https://www.lri.fr/hansen/cmatutorial.pdf>
  - 0.3: Randomized Black-Box optimization
  - 5: Discussion
- explain how the sampling works
- Consider the search costs together with the achieved objective function value.
- define *fitness function*

##### C. Constraints implementation

The CMA-ES source code handles box constraints natively using the "repair and penalize" policy. This consists in repairing non-feasible sampled points and penalizing (i.e. increasing the value of the fitness

function) proportionally to the distance between the unfeasible point and the search space. This policy forces the algorithm to eventually enter and stay in the feasible domain while avoiding evaluating non-allowed points.

As we will see, this policy is more widely applied in our method, in order to add linear constraints to the variables, reducing the size of the search space.

#### On single-knob groups:

These boundaries are 1-D box constraints, they are directly handled by the CMA-ES implementation as explained in IV-B.

#### On multiple-knob groups:

For the multiple-knob groups, the boundaries are refined using the same idea as in ?? applied to the *knob group flow demands*. The CMA-ES code requires however to implement ourselves a repair & penalize process instead of changing the knob boundaries, as the constraints will not be box but linear. (<- rephrase this).

For each multiple-knob group, we want to impose that the total flow brought by the knobs is in a range based on its *flow demand* and computed similarly to ??. The implementation is done by leaving the knob boundaries to their physical value (see ??) but *repairing* the points sampled by CMA-ES before imputing them to the simulator.

This reparation consists in a projection of the sampled point to the closest point on the feasible space, which is the intersection of the hypercube formed by the physical constraints and the volume between the two hyperplans defined by the range of the flow demands.(<- rewrite this)

#### Repairing:

Let  $\vec{k}^{(p)}$  the knobs vector sampled by CMA-ES at iteration p (i.e. before repairment).

Let  $J = \{j \in \llbracket 1, \gamma \rrbracket \mid \text{Card}(g_j > 1)\}$

$$\forall j \in J, \Delta_j^{\min} = \min \{\lambda \cdot \Delta_j; \Delta_j - I^{\text{local}}\}$$

$$\text{and } \Delta_j^{\max} = \max \{\Lambda \cdot \Delta_j; \Delta_j + I^{\text{local}}\}$$

The projection is implemented using the following program:

$$\begin{aligned} &\text{minimize} \quad \left\| \vec{k}^{(p)} - \vec{k} \right\|_2 \quad \text{<- Define norm 2 ?} \\ &\text{s.t.} \\ &\forall j \in J, \Delta_j^{\min} < \sum_{i \in g_j} \sigma_i \cdot k_i \cdot T_i < \Delta_j^{\max} \\ &\text{and } \vec{0} \leq \vec{k}^{(p)} \leq \vec{m} \end{aligned}$$

Fig. 2 above illustrates this projection and the two *tolerance hyper-plans* on a two-knobs group  $g_l = \{i, j\}$ , formed by on-ramp  $i$  and off-ramp  $j$ .

In this figure, the external square is the hyper-cube corresponding to the physical boundaries.  $D$  is the hyper-

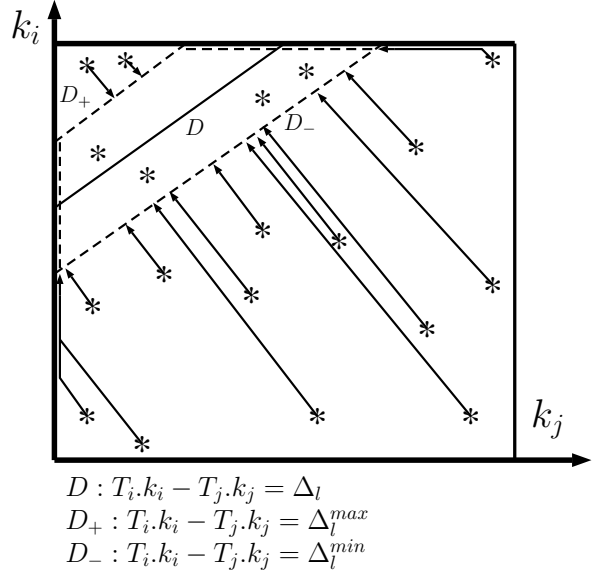


Fig. 2. Example of projection on a two-knobs group

plan (straight line) on which the knob group flow will have the exact value  $\Delta_l$ . The two tolerance hyper-plans on which the knob group flow value will be  $\Delta_l^{\max}$  and  $\Delta_l^{\min}$  are respectively  $D_+$  and  $D_-$ . The points sampled in the feasible space (inside the dotted line trapezium) remain untouched while the others are projected on the nearest point of the edge of the feasible space.

This repair ensures that only feasible values of the input are tested and that the algorithm will not get stuck in an unfeasible well of the physical boundaries hyper-cube.

#### Penalizing:

A penalization proportional to the distance between the projected and original points is added to the fitness function (see ??). This ensures that the algorithm will come closer to the feasible space at every iteration until eventually entering and staying inside it. This feature is important as it avoids two imbalances on the sampling:

- Testing more points on the edges than we should: if the algorithm is left sampling points far from the edges, without penalization, it will have no incentive to prefer sampling next to the feasible space than far. This is an obstacle from entering the feasible space, as all the points on the straight lines perpendicular to the edges would be equivalent. This would lead to a situation where it is common that too much (or all) of the points are sampled on the edges, CMA-ES converging far from the feasible space.(<- rewrite this).
- Imbalance between the edge points tested: the edge points which are the projection of more unfeasible hyper-cube points than others will be sampled unfairly more often.

*Example:* in Fig. 2, the points on the mediator of the segment formed by the intersections of  $D$  and

the square while inside the square are more numerous than the points on any other straight with same slope. The points on one of these straights and below the feasible space are all projected to the same point of the edge of the feasible space. Therefore, without penalization, the middle of the segment cited above would be sampled more often than the other points of its edge, for a reason that is not the simulation output it leads to.

The penalization due to the projection of the knobs in multiple-knob groups is proportional to the distance between the projected and originally sampled points. The normalizing factor is the distance between the maximums and minimums vectors, reflecting the *order of magnitude* of the search space.

$$\Pi_1(\vec{k}^{(p)}) = 100. \frac{\left\| \vec{k}^{(p)} - \underline{\vec{k}} \right\|_2}{\left\| \vec{k}^{max} - \vec{k}^{min} \right\|_2}$$

from tvn

This *a priori* calculation empowers us to project the input  $\vec{k}^{(p)}$  in order to ensure  $VMT(\vec{k}^{(p)}) \approx VMT^{PeMS}$ .

Where should the following part go?

Furthermore, in addition to the contribution of  $\Pi_2$ , a penalization proportional to the distance traveled by the projection is added to the fitness function. This implements a project & penalize process, for the same reasons as in ??.

Repairing:

The uncertainty  $I^{global}$  acts on this projection: the knobs vector will be projected on the space between 2 hyperplans of  $\mathcal{Q}$  defined by  $VMT^{PeMS} \cdot (1 - I^{global}) = VMT^{apriori}(\vec{k}^{(p)})$  and  $VMT^{PeMS} \cdot (1 + I^{global}) = VMT^{apriori}(\vec{k}^{(p)})$  and limited by the physical boundaries. The projection is done with the following program:

$$\text{minimize} \quad \left\| \vec{k}^{(p)} - \underline{\vec{k}} \right\|_2$$

<- define  $\vec{k}^{(p)}$ , clarify the three notations corresponding to the steps of the two input projections  
s.t.

$$\begin{aligned} VMT^{apriori}(\vec{k}^{(p)}) &> VMT^{PeMS} \cdot (1 - I^{global}) \\ VMT^{apriori}(\vec{k}^{(p)}) &< VMT^{PeMS} \cdot (1 + I^{global}) \\ \vec{k}^{(p)} &\in [\vec{0}, \vec{m}] \end{aligned}$$

Penalizing:

The penalization is proportional to the distance between the original point and projected point, normalized by the same *order of magnitude* as in ??.

## V. EXPERIMENT

### A. The traffic model : CTM

qs

### B. The large traffic simulator : BeATS

The macroscopic traffic simulator used is the Berkeley Advanced Traffic Simulator (BeATS). *talk about BeATS in "split ratio as output" mode*

The inputs of the simulator are:

- Fundamental Diagram of every link (4 parameters for each: *capacity*, *congestion speed*, *free-flow speed* and *jam density*).
- Entry flow of every source and sink. Timestep is 5 minutes, as for the data.

The outputs are the entry and exit flows, density and speed in every link.

### C. Context: Origin and reliability of the data

The real scenario used for this study is a segment of freeway 210 East in the suburbs of Los Angeles. The measurements used are collected by CalTRANS sensors and gathered in PeMS *talk about PeMS*.

These daily measurements are flow, density and speed on 74 links of 188, every 5 minutes for 24 hours.

After deleting partial or too biased data, the sensors have the following distribution:

- 33/135 monitored mainline links
- 26/28 monitored on-ramps
- 15/25 monitored off-ramps

*Should we introduce and quantify precisely the uncertainty notations  $I^{global}$  and  $I^{local}$  here or later ?*

*Will we talk about how we pre-processed the wrong sensors/ramp that comes from the left into the HOV, biased sensors etc. ?*

### D. Implementation of the congestion pattern

- Error computation from BeATS output:
  - A congestion threshold is defined for each mainline link :

$$\frac{\text{Link capacity}}{\text{Link freeflow speed}} + \delta$$

- The number of pixels of the contour plot that are in the wrong state is the error:

$$CP(p) = \sum_{t \in \frac{24h}{dt}} \sum_{l \in L} \mathbb{1}_{\text{wrong congestion state links}}$$

- Relative difference :  $\tau_p(CP) = 100 * \frac{CP(p)}{\text{Total area of the boxes}}$



### E. Parameters of the imputation

In our case, the total number of parameters taken by the simulator is 12890: 4 per fundamental diagram and 289 per source and off-ramp.

In this experiment, the fundamental diagrams have been calibrated using Gune's algorithm *talk about Gune's algorithm and the wide tolerance that allow FDS if they are more or less OK (Greg said).*

For each monitored source and off-ramp, the collected entry flow data is set as input.

The imputation therefore lays only on the entry flows of the non-monitored ramps.

In order to reduce the number of variables to one per non-monitored ramp, a custom flow *template* is built for each of these ramps. These templates are non-scaled daily flow profiles: each one of them as to be multiplied by a factor called *knob*, in order to correspond to the the objective. *talk about how the templates have been built.*

## VI. RESULTS

- 1) Effects of changing the parameters(initial standard deviation, changing the boundaries, changing the weights, changing the uncertainty, changing size of population).
- 2) Describe quality and usefulness of the result
- 3) Talk about uniqueness. Way to improve (test experiment): increasing population size. *Should we contact the creator of CMAES to ask him about the uniqueness of the solution (i.e. how to improve it ?*
- 4) Is our problem "noisy" ?
- 5) Talk about how cmaes behaves the way we want
- 6) Talk about what happens when we tune also the monitored ramps knobs.
- 7) Talk about issues:
  - a) Limit to result quality due to templates/FDS
  - b) Constraints handling has to be improved because several knobs end on their boundaries values
  - c) Uncertainties are symmetric: making them fit the sensors bias (e.g. : if they always under estimate) would be better.
- 8) BLABLABLA

## VII. CONCLUSION

- Multi-objective CMAES
- Find the knobs values that are a common good value for each day instead of the best for the average day
- Talk about tuning the FDS and Templates
- Talk about the uncertainty handling by Hansen and Niederberger,
- open to other stuff
- deepen understanding of "parameter sensitivity" and the adequation of having same scale for all knobs (and fds).
- BLABLA

## VIII. DEPRECATED PARAGRAPHS

If a group is single-knob, then the knob value is uniquely determined by this principle. We will call it *perfect value* of the knob.

If, on the contrary, it is a multiple-knob group, then the knobs of the group are linked by one linear equation. That is: their values are on a hyperplan of the space formed by their value as coordinates in an orthonormal base.

An uncertainty must be introduced on these flow demands due to the inexactness of both the templates and the simulator in general. This uncertainty consists in two multiplicative factors applied to the knob group flow demands. We call them *under-evaluation tolerance coefficient* and *over-evaluation tolerance coefficient*, denoted respectively  $\lambda$  and  $\Lambda$ . *see if we unify these two factors in one unique symmetric uncertainty*

A different uncertainty on the flow demands called *local uncertainty* has been introduced in I-C. As a remainder, it is caused by the uncertainty and bias on the measurements of the sensors. This uncertainty consists on a confidence interval centered in the value of the flow demand. We denote this local uncertainty  $I^{local}$ . As we will see just below, these two uncertainties compete to ensure a minimum width for the knob boundaries.

*Consider deleting this subsection and unifying the single-knob and multiple-knob groups into one unique description. There would be no need for perfect values, the single-knob group would be a particular case where no projection is needed.*

Defined only for single-knob groups, the *knob perfect values* are equal to the value of the corresponding knob group flow demand divided by the sum of the knobs template, in order to ensure that the flow going trough the corresponding ramp corresponds exactly to the flow demand.

*Notation:*

*Knob perfect values* :  $(k_i^*)_{i \in P}$   
with  $P = \{i \in K \mid \exists g \in G \text{ s.t. } g = \{i\}\}$

The perfect values are computed as follows, using that the total daily flow going trough a knob-ramp is the value of its knob times the sum of its template:

$\forall i \in P$ , given  $j$  s.t.  $g_j = \{i\}$  and  $T_i = \sum_{t=0}^{24h} t_i(t)$  :

$$k_i^* = \frac{\Delta_j}{T_i}$$

The boundaries of the knobs of single-knob groups are refined (<- *change this word*) as follows:

$$\begin{aligned} \text{Let } \forall i \in \llbracket 1, \kappa \rrbracket, \\ k_i^{\min} &= \max(\{\min(\{\Lambda \cdot k_i^*; k_i^* - \frac{I^{local}}{T_i}\}); 0\}) \\ k_i^{\max} &= \min(\{\max(\{\lambda \cdot k_i^*; k_i^* + \frac{I^{local}}{T_i}\}); m_i\}) \end{aligned}$$

Then denoting  $\vec{k}^{\min} = (k_1^{\min}, k_2^{\min}, \dots, k_\kappa^{\min})$  and  $\vec{k}^{\max} = (k_1^{\max}, k_2^{\max}, \dots, k_\kappa^{\max})$ , we impose :

$$\forall p \in \mathbb{N}^*, \quad k^{\min} \leq k^{(p)} \leq k^{\max}$$

The preceding formulas, defining the refined boundaries for single-knob groups, are the mathematical transcriptions of the two following steps:

- 1) For each knob and extremum, take the most permissive boundaries between what is obtained by multiplying the perfect value by the tolerance coefficients and what is obtained by adding/subtracting the local uncertainty.
- 2) Set all the extrema obtained in 1) that exceed the physical boundaries defined in ?? to their physical boundary value. (<- This is not clear)

This method allows us to quantify the freedom given to the result: the total daily flows of each non-monitored ramp is between  $\lambda$  and  $\Lambda$  times what has been measured by the mainline sensors, given that we accept a  $100 * I^{local}\%$  (<-depends on how  $I^{local}$  has been defined but this is the idea)uncertainty on these measures.

Taking into account  $I^{local}$  is indispensable as the computation of the perfect values leads sometimes to ridiculously small values. In these cases, the maximum obtained with  $\Lambda \cdot k_i^*$  corresponds often to a total daily flow of less than 50 cars exiting the ramp, which is not acceptable.

*Example:* One of the ramps has a perfect value of 0.02, which leads to a maximum of ?? cars going through the ramp during the whole day if  $\Lambda$  is set to 2 (very permissive: the daily flow can double what is measured by the mainline sensors). Once  $I^{local}=10\%$  is taken into account, the maximum of the knob becomes 0.7, which corresponds to ?? cars and is in an acceptable range.

#### ACKNOWLEDGMENT

#### APPENDIX

#### MODEL FIGURE